Improving the Convergence of Madhava-Gregory Series and a Rudimentary Calculation for π

Brian P. Pinto
Klude Pty Ltd, Adelaide, Australia
Published 13-March-2023

1. Abstract

The Madhava-Gregory series which is used in the Leibniz formula for π is said to converge slowly. This notion deems the formula less useful for practical applications rendering other formulas more prominent. But this same formula with no modification but a little change in perspective can make it quite competent with others. This paper shows how to improve the convergence of the series and make the formula fit for practical applications. Additionally, this paper also shows that a basic trigonometric limit identity, combined with an infinite precision in calculations, using basic arithmetic can be used to compute the value of π.

2. Introduction

During an attempt to build an experimental mathematical system, I needed to use an existing mathematical system that provided proofs for the relationships between the various formulas in the experimental system. I chose geometry and basic trigonometry to address the proofs that were needed. But I encountered an anomaly in the formulas of the experimental system when I tried to use them to derive the value of π. However, during this time, I learnt that the Madhava-Gregory series could be easily used to get the value of π in other ways than just using a single case where $x = 1$, which is nothing but the Leibniz formula for calculating value of $\frac{\pi}{4}$. While searching the internet, I was not able to find enough information on the matter of increasing the convergence of this series.

I believe that I have to justify the need to publish such a paper. And so, I wish to mention that while experimenting with various methods in the experimental system, I noticed that in whatever approaches I used, all values would cancel out, and eventually the Madhava-Gregory series would always turn up as the final equation for π. This resulted in instances where I could see a pattern in the change of convergence. But this information was not

available in any documents that I searched on the internet. Besides, at this point (at the time of writing of this document), all information that I can collect on the Madhava-Gregory series points to its inability to converge fast enough. So, I have proceeded to derive a proof, even though it is quite superfluous, and explain how to work with the convergence of this series. It should be noted that, though this proof is an obvious relationship, I did not arrive at the proof from the method I show in this document. Actually, I found solutions to various cases which resulted in the Madhava-Gregory series, and I saw a pattern from which I did an inductive proof. But the one I give here is a simple and straightforward mathematical relationship.

During the development of the formula to increase the convergence I also came up with a way to practically show realization of the trigonometric limit identity of $x\tan(\frac{\pi}{x})$. Using this, I show here that we can get a rudimentary algorithm that uses simple arithmetic to calculate the value of π.

3. The Madhava-Gregory series for $\frac{\pi}{4}$

The Leibniz Formula for π, which is actually widely known in the form of the value for $\frac{\pi}{4}$, is an infinite series for $\arctan(t)$ for $t = 1$ and is given as follows:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots \tag{1}$$

The underlying series is nothing but the infinite Taylor series of the inverse tangent function with the expansion at the origin, and is given by:

$$\arctan(t) = t - \frac{t^3}{3} + \frac{t^5}{5} - \frac{t^7}{7} + \frac{t^9}{9} - \frac{t^{11}}{11} + \ldots \tag{2}$$

which, in sigma notation is given as:

$$\arctan(t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} t^{2n+1} \tag{3}$$

When a value of 1 is plugged into the variable t the function yields a value of $\frac{\pi}{4}$.

$$\arctan(1) = 1 - \frac{1^3}{3} + \frac{1^5}{5} - \frac{1^7}{7} + \frac{1^9}{9} - \frac{1^{11}}{11} + \ldots \tag{4}$$

In trigonometry it is known by the integral:

$$\arctan(t) = \int_0^t \frac{1}{1+x^2}\, dx \tag{5}$$

This formula is a combination of Cavalieri's quadrature formula with Maclaurin series for $\arctan(x)$ [1] that gives the area under the arc of a circle and the series converges for $|t| \leq 1$.

One specific statement from the internet says:

> *Because of its exceedingly slow convergence (it takes five billion terms to obtain 10 correct decimal digits), the Leibniz formula is not a very effective practical method for computing $\frac{1}{4}\pi$. Finding ways to get around this slow convergence has been a subject of great mathematical interest.* [2]

This issue of convergence is already addressed in the series itself. The idea is to see what affects the convergence. It can be noticed that to increase convergence the value of t needs to decrease. Then all we need to do is find values of $t < 1$ that have a connection with π.

We already know what happens when $t = 1$ in (2). And we can easily verify the following using the series:

- $\frac{\pi}{6} = \arctan(\frac{1}{\sqrt{3}})$.
- $\frac{\pi}{12} = \arctan(2 - \sqrt{3})$.
- And go on ...

We can also see it diverging when you have:

- $\frac{\pi}{3} = \arctan(\sqrt{3})$

Therefore, we can conclude that for a result $\frac{\pi}{x}$ when x is 4 or greater, we can get the series to converge. Now you begin to see a way to prove it by induction. However, a simpler proof exists.

4. The proof for the value of π from the Madhava-Gregory series

We know the following from the definition of $\tan$ that if:

$$\tan\theta = t \tag{6}$$

then:

$$\theta = \arctan(t) \tag{7}$$

From the understanding of inverse of trigonometric functions we can say:

$$\theta = \arctan(\tan\theta) \tag{8}$$

Here, we can make θ a fraction of π by saying:

$$\frac{\pi}{K} = \arctan(\tan\frac{\pi}{K}) \tag{9}$$

We now bring in the Madhava-Gregory series (we will use the sigma notation):

$$\arctan(t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} t^{2n+1} \tag{10}$$

and replace $\arctan(t)$ and t with values from (9). What we get is:

$$\frac{\pi}{K} = \arctan(\tan\frac{\pi}{K}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} (\tan\frac{\pi}{K})^{2n+1} \tag{11}$$

Rearranging we get:

$$\boxed{\pi = K \sum_{n=0}^{\infty} \frac{(-1)^n (\tan\frac{\pi}{K})^{2n+1}}{2n+1} \quad (\text{where } K \geq 4)} \tag{12}$$

The equation (12) could become the most elegant computation for the value of π and the proof could be complete. But there is one issue. There is a π inside the R.H.S of the equation. We may not be able to write a clean equation without π on both sides - but we can look at K for the solution. However, before we address the issue, we will note a few points. They are as follows:

- In the equation (12) the value of K becomes our constant.
- Increase in K means decrease in the value of $\tan\frac{\pi}{K}$ which in turn increases the convergence of the series.
- It is important to note that we can't use just any arbitrary small value in place of $\tan\frac{\pi}{K}$. Because it must be linked with K. This is because we need K to be '*multiplied with*' after the series is summed up.
- We can say $T = \tan\frac{\pi}{K}$, making T another interlinked constant.
- Both K and T need to be obtained and they need to be linked.

- In practical applications obtaining K is a challenge. This is because from what I could collect, there may not be a direct formula to calculate the value of $\tan\frac{\pi}{K}$. What we have is one for $\tan t$[1] which is a series in itself:

$$\tan(t) = t + \frac{t^3}{3} + \frac{2t^5}{15} + \frac{17t^7}{315} + \frac{62t^9}{2835} + \dots \tag{13}$$

  But then we have to use the value of π to begin with, and insert it into the formula to get the value of π. So, we are back to square one.

- We don't need a specific value of t, we only need a value of t such that t is very tiny, and the associated angle has a form $\frac{\pi}{K}$, and we need both K and T.

5. Algorithm for π - Obtaining K and T

There may be a more efficient method to get the value of K by using a straightforward mathematical equation (but I am unaware of it). I have tried various methods, unfortunately, all ending up with nested square roots. So, we need to deal with nested square roots. But because square roots take up a lot of processing time, we need to minimize the operations to get an efficient algorithm. In this paper I will provide an iterative programmatic method I used, during the *value of π proofing* experiments, to obtain K with basic mathematical operations.

The idea is to select one angle that is a ratio of π for which we know the value of its inverse trigonometric function. With this as our starting point we keep dividing the angle using half angle formulas. If we keep track of the number of times we divide the angle, that will contribute to the K, in the course we get the value of T as well. Then we use these as constants in (12) and choose the number of series iterations to suit our precision of π. Here is how we do it:

We know that the two trigonometric functions of a fundamental commonly used angle $\frac{\pi}{3}$ results in values as follows:

$$\cos(\frac{\pi}{3}) = \frac{1}{2} \tag{14}$$

$$\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2} \tag{15}$$

We are also aware of the trigonometric half-angle identities for $\cos\theta$ and $\tan\theta$ as follows:

$$tan\frac{\theta}{2} = \sqrt{\frac{1-\cos\theta}{1+\cos\theta}} \tag{16}$$

$$cos\frac{\theta}{2} = \sqrt{\frac{1+\cos\theta}{2}} \tag{17}$$

We see that $\cos$ connects with $\tan$ very well in the half angle suite. We can now use $\cos(\frac{\pi}{3}) = \frac{1}{2}$ as a starting point and create an algorithm and iteratively divide the angle as we go. Our equation from (12) looks as follows:

$$\pi = K\sum_{n=0}^{\infty}\frac{(-1)^n(T)^{2n+1}}{2n+1} \tag{18}$$

We need the values of K and T so we can plug it into the series to get P which is the value of π. $R+1$ will be the number of times we divide the angle. This we term as the *TAN-DIP*. L will be then number of iterations we run on the series. The algorithm goes as follows:

$$
\begin{aligned}
&\text{Start TAN-DIP iteration with initial value of } 1/2 \\
&v_1 = (\frac{1}{2}) \\
&v_2 = \sqrt{\frac{1+v_1}{2}} \\
&v_3 = \sqrt{\frac{1+v_2}{2}} \\
&. \\
&. \\
&. \\
&v_R = \sqrt{\frac{1+v_{R-1}}{2}}
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
&\text{After the TAN-DIP iterations, get the value of tan} \\
&T = \sqrt{\frac{1-v_R}{1+v_R}}
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
&\text{Determine K} \\
&K = 3 \times 2^{R+1}
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
&\text{Put the value into the series} \\
&P = K\sum_{n=0}^{L} \frac{(-1)^n (T)^{2n+1}}{2n+1}
\end{aligned} \tag{22}
$$

Following are some of the observations on this algorithm:

- We can choose the starting point from $\frac{1}{\sqrt{2}} = \cos\frac{\pi}{4}$. Then we have to change (21) as $K = 4 \times 2^{R+1}$. But we need to make sure that we do at least one of the $\cos$ half-angle iterations before we do the $\tan$ half-angle calculation. Any starting point below $1 = \cos\frac{\pi}{2}$ can be used with the right initial numbers.
- We cannot choose starting point from $1 = \cos\frac{\pi}{2}$. Which is obvious.
- If $L = \infty$ and we have infinite precision the $P = \pi$.
- K is proportional to 2^{R+1} therefore increasing the TAN-DIP increases the convergence. In other words, higher the TAN-DIP, smaller the angle, greater the K, smaller the T, steeper the convergence.

- Once K and T are calculated for a very high precision and a large TAN-DIP, it takes very few iterations for the Madhava-Gregory series to converge to admirable precision.
- TAN-DIP iterations are more processor intensive because of the square root calculations in each iteration.
- The algorithm is in its pure form without any numerical adjustments, directly connected to the basic trigonometric identities and power-series functions.
- Any adjustments visible in this algorithm like precision and iteration control are for the sake of practicalities.

6. The Rudimentary Algorithm for π

There is one more special observation that is worthy of a separate mention. If we take the algorithm at (22) and increase the TAN-DIP to a very high value, the convergence goes up and it helps us to use lesser and lesser iterations of the series to get the value of π. We can even reduce the iteration to just 1 and still get a decent number of digits to match. This stems from the trigonometric limit identity[5] given as:

$$\lim_{n \to \infty} \left(n \tan \frac{\pi}{n} \right) = \pi \tag{23}$$

This can be seen not just on paper but in programming reality and it is amazing to witness. In the above algorithm:

- K in (21) is the n of (23)
- T in (20) is the $\tan \frac{\pi}{n}$ of (23)

Then equation (23) becomes:

$$\lim_{K \to \infty} (KT) = \pi \tag{24}$$

So if we get very super-high precision, super-high TAN-DIP K and T and just multiply as below we get a decent value of π:

$$P = KT \tag{25}$$

This makes the whole algorithm shrink to:

$$v_1 = (\frac{1}{2})$$
$$v_2 = \sqrt{\frac{1+v_1}{2}}$$
$$v_3 = \sqrt{\frac{1+v_2}{2}}$$
$$.$$
$$.$$
$$.$$
$$v_R = \sqrt{\frac{1+v_{R-1}}{2}}$$
$$T = \sqrt{\frac{1-v_R}{1+v_R}}$$
$$K = 3 \times 2^{R+1}$$
$$P = KT$$

!!! NO SERIES REQUIRED !!!

Following are some of the observations on this algorithm:

- K is proportional to 2^{R+1} therefore increasing the TAN-DIP increases the precision of π digits.
- The algorithm is in its purely arithmetic form and without any numeric adjustments.
- Related formulas which are close to this one are Viète's formula[6] and Liu Hui's π algorithm [7]. But both of them are of geometric or geometric-trigonometry origin. This algorithm is of the trigonometric-calculus origin.

7. Experiments and Results

A simple code with low TAN-DIP, is written in PHP and shown below. With low TAN-DIP it doesn't amount to much. TAN-DIP calculations take a lot of time as well.

```
$R = 10;
$iterations = 5;
$pi_decimals = 2000;
$precision = $pi_decimals*3;
bcscale($precision);
```

```
// TAN DIP
$v = bcdiv(1,2);
$j = $R - 1;
for ($i=0; $i < $j; $i++) {
    $v = bcsqrt(
        bcdiv(bcadd(1,$v),2)
    );
}
$K = bcmul(3, bcpow(2, $R));
$T = bcsqrt(bcdiv(bcsub(1,$v),bcadd(1,$v)));

// THE SERIES
$sum = 0;
for ($i =0; $i < $iterations; $i++) {
    $exp = 1 + (2 * $i);
    $sum = bcadd(
        $sum,
        bcdiv(
            bcmul(pow(-1, $i), bcpow($T, $exp)),
            $exp
        )
    );
}
$P = bcmul($K, $sum);
```

Snapshots of the results are shown below, they are very modest because the convergence is slow due to low TAN-DIP and the total series iterations are controlled to 5. It managed to match 31 digits.

actual	3.141592653589793238462643383279502884197169399375105820974
matched	**3.141592653589793238462643383279**860194123957232143013882393
matched_digits	31
precision	6000
iterations	5
R	10
T	0.001022654215094841154940098025921773462183801600676542003
K	3072
tandip-time	15978mS
series-time	452mS

Figure 1: Simple Algo Sample Result

During the development for this paper multiple experiments were carried out with various precisions and TAN-DIPs. These were done as a test to see what the effects were of various parameters like K, T, L, R etc. But they were not designed to profile or measure the performance as it was not a part of the study. During these experiments the following was observed:

- TAN-DIP needs to be at least 1000 to get good admirable results and the calculations must be done with at least 3 times the precision required.
- On many experiments TAN-DIP of 1001 with precision 6000 (K-T-6000-1001) was used as a constant.
- With K-T-6000-1001 as a constant, π-precision even up to 3000 digits could be got with 5 iterations.
- Some examples are shown below:

actual	3.1415926535897932384626433832795028841971693993751058209749
matched	**3.1415926535897932384626433832795028841971693993751058209749**
matched_digits	3014
iterations	5
R	1001
V	0.0000000000000000000000000000000000000000000000000000000000
K	6429051643117603925690550294360010863368428870233201644662
series-time	449mS

Figure 2: With large R and high precision but pre calculated constant.

actual	3.1415926535897932384626433832795028841971693993751058209749
matched	**3.1415926535897932384626433832795028841971693993751058209749**
matched_digits	603
iterations	0
R	1001
V	0.0000000000000000000000000000000000000000000000000000000000
K	6429051643117603925690550294360010863368428870233201644662
algo	2mS

Figure 3: An example of just multiplying K with T with large R and high precision.

actual	3.1415926535897932384626433832795028841971693993751058209749
matched	**3.1415926535897932384626433832795028841971693993751058209749**
matched_digits	2037
iterations	146
algo	2969mS

Figure 4: An example of Chudnovsky's algorithm based results
(Algorithm - Courtesy of https://github.com/natmchugh/pi)

Some points to note:

- This paper deals with the proofs of the Madhava-Gregory series and how to deal with convergence and get results but not the quality of results.
- The various practical aspects such as profiling, and performance of this algorithm are not the focus of the paper. The experiments to study the effects of variations in the parameters such as K, T, L, R etc. must be done separately.
- All experiments were done in `PHP 7.3.4 on a Windows 10 PC with Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 24.0 GB RAM`.

8. Conclusion

These conclusions are voiced as opinions and no more:

- The Madhava-Gregory series is a natural (free of numeric adjustments) method to calculate the value of π.
- Moreover, because this formula has its basis in straightforward simple trigonometry, it can be the choice to provide the simplest of proofs for the value of π.
- In terms of the algorithm we have seen that convergence could be improved using (K, T) which is an interlinked constant pair.
- The profiling, performance, radius of convergence, parametric analysis etc., of the algorithm must be a part of another study.
- If calculation of (K,T) is not considered as part of this algorithm, then this algorithm, with a good (K,T), performs superior to the existing Chudnovsky's algorithm. But such a claim would be wrong as the pair (K,T) itself is part of the calculation and the time taken by the algorithm, because the pair is part of the calculation of π. Chudnovsky's algorithm is currently still the best at speed.
- In terms of the *Spirit of Mathematics*, just as we have rudimentary geometric calculations for the value of π through iterations of polygons, we also see that a rudimentary calculation for π in terms of pure arithmetic exists through the knowledge of the trigonometric-calculus.

9. Notes and References

- [1] Boyer, Carl B.; Merzbach, Uta C. (1989) [1968]. A History of Mathematics (2nd ed.). Wiley. pp. 428–429.
- [2] Gregory's series - Wikipedia https://en.wikipedia.org/wiki/Gregory%27s_series
- [3] Roy, Ranjan (1990). "The Discovery of the Series Formula for π by Leibniz, Gregory and Nilakantha". Mathematics Magazine. 63 (5): 291–306. doi:10.1080/0025570X.1990.11977541
- [4] Power Series Expansion for Tangent Function https://proofwiki.org/wiki/Power_Series_Expansion_for_Tangent_Function
- [5] Power Series Expansion for Tangent Function https://math.stackexchange.com/questions/3224455/if-lim-n-tan-frac180n-pi-is-infty-times0-pi
- [6] Viète's formula - Wikipedia https://en.wikipedia.org/wiki/Vi%C3%A8te%27s_formula
- [7] Liu Hui's π algorithm - Wikipedia https://en.wikipedia.org/wiki/Liu_Hui%27s_%CF%80_algorithm